IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application No.: 10/091,176	§	Examiner: Harrison, Chante
Filed: March 5, 2002	§	Group/Art Unit: 2677
Inventor(s):	§	Atty. Dkt. No: 5181-84600
Charles Patton	§	
	§	
	§	
Title: PIPELINED 2D VIEWPORT CLIP CIRCUIT	§	
	§	
	§	
	§	

APPEAL BRIEF

Sir/Madam:

Further to the Notice of Appeal filed February 26, 2007, Appellant presents this Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by SUN MICROSYSTEMS, INC., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 901 San Antonio Road, Palo Alto, CA 94303, as evidenced by the assignment recorded at Reel 013061, Frame 0090.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 18-40 are pending and rejected. The rejection of claims 18-40 is being appealed. A copy of claims 18-40 is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 18 is directed to a method for comparing a pixel location against a plurality of windows. The method comprises passing the pixel location through a plurality of clip circuits, wherein the clip circuits are connected in a series to form a pipeline, and wherein each clip circuit is a segment of the pipeline (*as disclosed at least at page 5, lines 3-8 (as amended 2/23/2005), and at page 19, lines 1-8, of the specification*).

The method further comprises computing a window result in each clip circuit for the pixel location, wherein each clip circuit is provided data defining a different one of the plurality of windows, wherein the window result comprises an indication of inclusion of the pixel location within the corresponding one of the plurality of windows (*as disclosed at least at page 19, lines 9-13, of the specification*).

The method further comprises outputting the pixel location and a window word from each clip circuit, wherein said outputting comprises, passing the pixel location and the window word directly to a next clip circuit in the series of clip circuits except for the last clip circuit in the series, and wherein the window word also comprises any previous window results (*as disclosed at least at page 19, lines 14-19, of the specification*).

The method further comprises examining the window word output by the last clip circuit in the series of clip circuits to determine if the pixel is included in at least one of the windows (*as disclosed at least at page 19, lines 20-22, of the specification*).

Independent claim 26 is directed to a method for comparing a pixel location against a plurality of windows. The method comprises supplying window boundary coordinates for a different one of a plurality of windows to each clip circuit of a plurality of clip circuits connected in a series (*as disclosed at least at page 5, lines 3-8 (as amended 2/23/2005), and at page 19, lines 1-8, of the specification*).

The method further comprises determining inclusion of a pixel in the corresponding window in each clip circuit (*as disclosed at least at page 19, lines 9-13, of the specification*).

The method further comprises passing the pixel and a result of said determining inclusion to a next clip circuit in the series of clip circuits, except for a last clip circuit of the series of clip circuits (*as disclosed at least at page 19, lines 14-19, of the specification*).

Independent claim 34 is directed to a system for determining inclusion of a pixel with respect to each of a plurality of windows. The system comprises a plurality of clip circuits connected in a series (*as disclosed at least at page 5, lines 3-8 (as amended 2/23/2005), and at page 20, lines 20-23, of the specification*).

Each circuit in the series is configured to: (a) receive horizontal and vertical coordinates locating a pixel, (b) receive horizontal and vertical coordinates defining a different one of a plurality of windows, and (c) compute a window result indicating inclusion of the pixel within the corresponding window defined in (b) (*as disclosed at least at page 21, lines 1-7, of the specification*).

Each circuit in the series except for a last clip circuit in the series is also configured to, (d) pass the horizontal and vertical coordinates of the pixel, the window result computed in (c), and any prior window results to a next clip circuit in the series of clip circuits (*as disclosed at least at page 19, lines 14-19, and at page 21, lines 1-7, of the specification*).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 18-40 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Takeda (U.S. Pat. No. 5,559,937), and further in view of Debra Kipping et al. (U.S. Pat. No. 6,831,660).

VII. ARGUMENT

Ground of Rejection:

Claims 18-40 are finally rejected in Office Action #6 dated 11/24/2006 under 35 U.S.C. § 103(a) as being unpatentable over Takeda (U.S. Pat. No. 5,559,937), and further in view of Debra Kipping et al. (U.S. Pat. No. 6,831,660) (hereinafter "Kipping"). Appellant traverses this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 18-24

Claim 18 recites: A method for comparing a pixel location against a plurality of windows, the method comprising:

passing the pixel location through a plurality of clip circuits, wherein the clip circuits are connected in a series to form a pipeline, and wherein each clip circuit is a segment of the pipeline;

computing a window result in each clip circuit for the pixel location, wherein each clip circuit is provided data defining a different one of the plurality of windows, wherein the window result comprises an indication of inclusion of the pixel location within the corresponding one of the plurality of windows;

outputting the pixel location and a window word from each clip circuit, wherein said outputting comprises, passing the pixel location and the window word directly to a next clip circuit in the series of clip circuits except for the last clip circuit in the series, and wherein the window word also comprises any previous window results; and

examining the window word output by the last clip circuit in the series of clip circuits to determine if the pixel is included in at least one of the windows.

Neither Takeda nor Kipping either singly or in combination teach or render obvious at least the following: "passing the pixel location through a plurality of clip circuits, wherein the clip circuits are connected in a series to form a pipeline, and wherein each clip circuit is a segment of the pipeline; computing a window result in each clip circuit for the pixel location, wherein each clip circuit is provided data defining a different one of the plurality of windows, wherein the window result comprises an indication of inclusion of the pixel location within the corresponding one of the plurality of windows".

The Examiner states at page 3 of Office Action #6, that "*Takeda discloses passing the pixel location through a plurality of clip circuits (i.e. passing polygon vertex data, col.8, ll.39-44; col. 20, ll. 59-65, through clip circuits Fig. 6c "420 a & b")...*". However, the cited passages of Takeda do not disclose passing **pixel** data, but rather teach passing polygon **vertex** data. Pixel and vertex are not equivalent terms in computer graphics. A pixel is a location in screen space that is to be rendered if found to be within

a defined "window" (i.e. visible in screen space). Vertices are the defining points of a polygon that corresponds to a portion of an object in object space.

The Examiner also states at page 3 of Office Action #6, that "*Takeda discloses ... wherein the clip circuits are connected in series to form a pipeline (Fig. 6c "420 a & b" connected in series col. 11, ll.30-35), and wherein each clip circuit is a segment of the pipeline (Fig. 6c "420 a & b" are segments of pipeline "410"); computing a window result (i.e. inside/outside decision stored as specification data) in each clip circuit for the pixel location...*". However, the two clipping circuits (420A and 420B of Fig. 6) that are connected in series **do not have the capability** of "computing a window result in each clip circuit for the pixel location….wherein the window result comprises an indication of inclusion of the pixel location within the corresponding one of the plurality of windows". Takeda teaches that circuit 420A (or 420B) is only capable of testing against 3 planes (of the 6 planes that define a window region) at col. 11, lines 25-32:

> *"An example of the above described processing by using a plurality of the clipping processing device is shown in FIG. 6C. In this case, for example, clipping processing by a first three clipping surfaces is performed by a **first-stage** clipping processing device 420a, and clipping processing by the remaining three surfaces and perspective projection conversion is performed by a **second-stage** clipping processing device 420b."*

However, Takeda clearly states that an inclusion decision requires testing against 6 planes at col. 2, line 43 through col. 3, line 25:

> *"Next, a type of image processing that is called clipping processing is performed. Clipping processing is image processing whereby image information that is outside the field of view of the player 302 (or outside the field of view of a window opening into the 3D space), in other words, **image information that is outside a region bounded by clipping surfaces 1, 2, 3, 4, 5, and 6 (hereinafter called a display region 20), is excluded**…….This is described below in more detail with reference to FIG. 19. Image information on an object outside the field of view of the player 302 (outside the display region 20), such as the 3D object 334 representing a billboard that has moved out of the field of view and backwards, is excluded. This exclusion processing is performed by **determining whether or not***

an object is within a display region for each of the clipping surfaces 1 to 6, then excluding the object only if it is outside all of those surfaces*.....However, as shown in FIG. 19, this clipping processing* ***has to be performed for all of the clipping surfaces 1 to 6****, and in practice the image processing regulates the speed of the entire circuitry of this 3D calculation section 316 the most."*

Clearly, the "first stage" circuit 420a and the "second stage" circuit 420b as taught by Takeda **must be** combined to **correspond to a single clip circuit of Appellant's claim 18**, since each circuit 420a or 420b only performs one half of the computation required to determine inclusion within a window region. Therefore, a "first stage" circuit 420a alone or a "second stage" circuit 420b alone as taught by Takeda is not capable of "computing a window result in each clip circuit for the pixel location....wherein the window result comprises an indication of inclusion of the pixel location within the corresponding one of the plurality of windows".

Consequently, since at least the combination of circuits 420a and 420b is required to correspond to one clip circuit as presented in claim 18, it is obvious that Takeda does not teach "passing the pixel location through a plurality of clip circuits, wherein the clip circuits are connected in a series to form a pipeline, and wherein each clip circuit is a segment of the pipeline; computing a window result in each clip circuit for the pixel location, wherein each clip circuit is provided data defining a different one of the plurality of windows, wherein the window result comprises an indication of inclusion of the pixel location within the corresponding one of the plurality of windows".

The Examiner in a prior Office Action also cited Kipping's element 310 of Fig. 3 as teaching clip circuits connected in a series. However, element 310 only shows a box with an internal label "Hardware Clippers 320", and therefore does not indicate any connection between Hardware Clippers 320. The Examiner also cited Kipping as teaching Hardware Clippers 320 connected in a series at col. 1, lines 45-68:

"With the large amounts of data and computations involved in processing graphics data, especially with three-dimensional applications, many of these computations have been offloaded from the central processing units to a graphics adapter. Within these graphics systems, a graphics pipeline located in the

graphics adapter is used to process this graphics data. With a pipeline, the graphics data processing is partitioned into stages of processing elements in which processing data may be executed sequentially by separate processing elements.

Within these processing elements, a clipping function is typically implemented in which pixels for a primitive located within a clip area are rendered, while pixels for the primitive outside of the clip area are not rendered. The hardware implementing the clipping function is often referred to as a hardware clipper. Often, more than one clip area is used to render the correct pixels on a display. With some high-end graphics adapters, multiple hardware clippers are present to clip graphics primitives. Clipping primitives, such as lines and segments, are faster using hardware clippers, rather than software clippers."

It is clear from these two paragraphs that Kipping teaches that processing data may be executed sequentially by separate processing elements, that a clipping function is one of the processing elements, and that the clipping function may use multiple hardware clippers. However, there is no teaching that multiple hardware clippers are connected in a series, nor is there any teaching of each hardware clipper determining a result for a corresponding window of a set of multiple windows. Therefore, Kipping teaches that a plurality of separate processing elements form a pipeline, but Kipping is silent on whether a single processing element such as the clipping function performs the clipping function using multiple clip circuits that are connected in a series to form a pipeline.

In addition, since there is no teaching in Kipping to connect a plurality of clip circuits in a series to form a pipeline, then clearly Kipping **also** fails to teach "passing the pixel location through a plurality of clip circuits, wherein the clip circuits are connected in a series to form a pipeline, and wherein each clip circuit is a segment of the pipeline; computing a window result in each clip circuit for the pixel location, wherein each clip circuit is provided data defining a different one of the plurality of windows, wherein the window result comprises an indication of inclusion of the pixel location within the corresponding one of the plurality of windows".

Furthermore, Takeda and Kipping either singly or in combination do not teach or render obvious "outputting the pixel location and a window word from each clip circuit, wherein said outputting comprises, passing the pixel location and the window word directly to a next clip circuit in the series of clip circuits". In addition, Takeda and Kipping either singly or in combination do not teach or render obvious "the window word also comprises any previous window results".

The Examiner states at page 4 of Office Action #6, that Takeda discloses *"outputting comprises, passing the pixel location and the window word directly to a next clip circuit in the series of clip circuits (Fig. 6c; col. 21, ll. 25-30; col. 22, ll. 25-32), except for the last clip circuit in the series (col. 22, ll. 32-35), and wherein the window word also comprises any previous window results…(col. 11, ll. 20-35; Fig. 6c; col. 22, ll. 25-30), and examining the window word output by the last clip circuit in the series of clip circuits to determine if the pixel is included in at least one of the display regions (col. 20-21, ll. 65-32)"*. However, the cited passages of Takeda do not disclose passing **pixel** data, but rather teach passing polygon **vertex** data. Pixel and vertex are not equivalent terms in computer graphics. A pixel is a location in screen space that is to be rendered if found to be within a defined "window" in screen space (i.e. visible in screen space). Vertexes are the defining points of a polygon that describe a portion of an object in object space. More specifically, Takeda teaches a clipping process for a polygon that intersects a 3D clipping surface. If there is more than one surface to process, then the second surface parameters and the vertexes for the clipped polygon from the first clipping are fed back first to an "**input section**" and then to the **same clip circuit** and the previously clipped polygon is tested for an intersection with the second surface (see Takeda col. 22, ll. 25-35). The cited passages refer to clipping polygons, not a pixel's location in screen space. Therefore, Takeda does not disclose: "passing the **pixel** location and the window word **directly** to a **next** clip circuit in the **series** of clip circuits", since there is no teaching in Takeda of passing a pixel and the window word directly to a next clip circuit, or in fact, since there is no next circuit; "the window word also comprises any previous window results", since a clipped polygon will not contain information on **any** prior clipping other than the combined result of all prior clippings of the polygon; "determine

if the pixel is included in at least one of the display regions", since there is no pixel in Takeda's teachings.

The Examiner in a prior Office Action also cited Kipping as teaching "outputting the pixel location and a window word from each clip circuit, wherein said outputting comprises, passing the pixel location and the window word directly to a next clip circuit in the series of clip circuits" at col. 4, lines 52-67:

> *"Clip buffer 316 contains clip planes, which are used as a mask to clip graphic primitives before the raster stores into the frame buffer 310. Using hardware clip planes is not as fast as the hardware clippers, but it is usually faster than clipping the graphic primitives in software. Raster engine 310 includes hardware clippers 320. A hardware clipper processes graphics data to determine whether the graphics data should be displayed. Before a pixel is written to a frame buffer, such as color frame buffer 312, the location of the pixel is compared to the region extents of the region set for the window clipper. The comparison determines if the location of the pixel is outside, on, or inside the region. In these examples, the pixel is not written to the frame buffer if the address is not contained on or inside the draw region or if the pixel lies on or inside a no draw region."*

Clearly, there is no teaching in this paragraph of Kipping of a series of clip circuits, nor of passing the pixel location and a window word directly to a next clip circuit in the series, nor that the window word passed between a clip circuit and the next clip circuit also comprises any previous window results. In fact, there is no teaching in Kipping of one clip circuit sending any information to another clip circuit.

Therefore, Appellant submits that claim 18 and its dependent claims are non-obvious and patentably distinguished over Takeda and Kipping for at least the reasons given above.

Claim 25

Claim 25 adds the limitation that the "plurality of clip circuits are identical circuits" to the limitations of claim 18 discussed above. Neither Takeda nor Kipping has any teaching on identical clip circuits in a series. For this reason and the reasons cited above for claim 18, Appellant submits that claim 25 is non-obvious and patentably distinguished over Takeda and Kipping.

Claims 26-33

Neither Takeda nor Kipping either singly or in combination teach or render obvious the limitations of claim 26: "*A method comprising: supplying window boundary coordinates for a different one of a plurality of windows to each clip circuit of a plurality of clip circuits connected in a series; determining inclusion of a pixel in the corresponding window in each clip circuit; and passing the pixel and a result of said determining inclusion to a next clip circuit in the series of clip circuits, except for a last clip circuit of the series of clip circuits*".

Appellant submits that claim 26 and its dependent claims are non-obvious and patentably distinguished over Takeda and Kipping for at least the reasons given above in support of claim 18.

Claims 34-40

Neither Takeda nor Kipping either singly or in combination teach or render obvious the limitations of claim 34: "*A system for determining inclusion of a pixel with respect to each of a plurality of windows, the system comprising: a plurality of clip circuits connected in a series, wherein each circuit in the series is configured to: (a) receive horizontal and vertical coordinates locating a pixel, (b) receive horizontal and vertical coordinates defining a different one of a plurality of windows, (c) compute a*

window result indicating inclusion of the pixel within the corresponding window defined in (b), and except for a last clip circuit in the series, (d) pass the horizontal and vertical coordinates of the pixel, the window result computed in (c), and any prior window results to a next clip circuit in the series of clip circuits".

Appellant submits that claim 34 and its dependent claims are non-obvious and patentably distinguished over Takeda and Kipping for at least the reasons given above in support of claim 18.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 18-40 was erroneous, and reversal of his decision is respectfully requested.

The fee of $500.00 for filing this Appeal Brief is being paid concurrently via EFS-Web. If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5181-84600/JCH.

Respectfully submitted,

/Jeffrey C. Hood/
Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: April 26, 2007 JCH/JWC

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. - 17. (Cancelled).

18. (Previously Presented) A method for comparing a pixel location against a plurality of windows, the method comprising:

passing the pixel location through a plurality of clip circuits, wherein the clip circuits are connected in a series to form a pipeline, and wherein each clip circuit is a segment of the pipeline;

computing a window result in each clip circuit for the pixel location, wherein each clip circuit is provided data defining a different one of the plurality of windows, wherein the window result comprises an indication of inclusion of the pixel location within the corresponding one of the plurality of windows;

outputting the pixel location and a window word from each clip circuit, wherein said outputting comprises, passing the pixel location and the window word directly to a next clip circuit in the series of clip circuits except for the last clip circuit in the series, and wherein the window word also comprises any previous window results; and

examining the window word output by the last clip circuit in the series of clip circuits to determine if the pixel is included in at least one of the windows.

19. (Previously Presented) The method of claim 18, wherein said pixel location comprises a horizontal and a vertical coordinate that define position of said pixel on a screen.

20. (Previously Presented) The method of claim 19, wherein each of the plurality of windows comprises a first horizontal and a second horizontal coordinate and a

first vertical and a second vertical coordinate that define each window's boundaries on the screen.

21. (Previously Presented) The method of claim 20, wherein said computing window result comprises:

computing horizontal inclusion by computing if said horizontal pixel coordinate is located between the first horizontal and the second horizontal coordinate of the corresponding window; and

computing vertical inclusion if said vertical pixel coordinate is located between the first vertical and the second vertical coordinate of the corresponding window.

22. (Previously Presented) The method of claim 21, wherein said computing window result further comprises: setting the indication of inclusion of the pixel to positive if both the horizontal and vertical inclusions are true, and setting the indication of inclusion of the pixel negative if one or more of the horizontal and vertical inclusions are false.

23. (Previously Presented) The method of claim 18, further comprising:

clipping the pixel if said examining determines that the pixel is not included in any of the plurality of windows; and

propagating the pixel if said examining determines that the pixel is included in at least one of the plurality of windows.

24. (Previously Presented) The method of claim 18, wherein said plurality of windows comprise two or more 2-D windows.

25. (Previously Presented) The method of claim 18, wherein said plurality of clip circuits are identical circuits.

26. (Previously Presented) A method comprising:

supplying window boundary coordinates for a different one of a plurality of windows to each clip circuit of a plurality of clip circuits connected in a series;

determining inclusion of a pixel in the corresponding window in each clip circuit; and

passing the pixel and a result of said determining inclusion to a next clip circuit in the series of clip circuits, except for a last clip circuit of the series of clip circuits.

27. (Previously Presented) The method of claim 26, wherein said pixel comprises a horizontal and a vertical coordinate that define position of said pixel on a screen.

28. (Previously Presented) The method of claim 27, wherein each of the plurality of windows comprises a first horizontal and a second horizontal coordinate and a first vertical and a second vertical coordinate that define boundaries of each of the plurality of windows on the screen.

29. (Previously Presented) The method of claim 28, wherein said computing window result comprises:

computing horizontal inclusion by computing if said horizontal pixel coordinate is located between the first horizontal and the second horizontal coordinate of the corresponding window; and

computing vertical inclusion if said vertical pixel coordinate is located between the first vertical and the second vertical coordinate of the corresponding window.

30. (Previously Presented) The method of claim 29, wherein said computing window result further comprises: setting the indication of inclusion of the pixel to positive if both the horizontal and vertical inclusions are true, and setting the indication of inclusion of the pixel negative if one or more of the horizontal and vertical inclusions are false.

31. (Previously Presented) The method of claim 26, further comprising:

clipping the pixel if said examining determines that the pixel is not included in any of the plurality of windows; and

propagating the pixel if said examining determines that the pixel is included in at least one of the plurality of windows.

32. (Previously Presented) The method of claim 26, wherein said plurality of windows comprise two or more 2-D windows.

33. (Previously Presented) The method of claim 26, wherein said plurality of clip circuits are identical circuits.

34. (Previously Presented) A system for determining inclusion of a pixel with respect to each of a plurality of windows, the system comprising:

a plurality of clip circuits connected in a series, wherein each circuit in the series is configured to:

(a) receive horizontal and vertical coordinates locating a pixel,

(b) receive horizontal and vertical coordinates defining a different one of a plurality of windows,

(c) compute a window result indicating inclusion of the pixel within the corresponding window defined in (b), and

except for a last clip circuit in the series,

(d) pass the horizontal and vertical coordinates of the pixel, the window result computed in (c), and any prior window results to a next clip circuit in the series of clip circuits.

35. (Previously Presented) The system of claim 34, wherein a first horizontal and a second horizontal coordinate and a first vertical and a second vertical coordinate corresponding to each window define boundaries of each of the plurality of windows in a two-dimensional space.

36. (Previously Presented) The system of claim 34, wherein the system is further configured to:

clip the pixel if all window results indicate the pixel is not included in any of the plurality of windows; or

propagate the pixel if all window results indicate the pixel is included in at least one of the plurality of windows.

37. (Previously Presented) The system of claim 34, wherein the plurality of windows comprises two or more 2-D windows.

38. (Previously Presented) The system of claim 34, wherein the plurality of clip circuits are identical circuits.

39. (Previously Presented) The system of claim 34, wherein each clip circuit of the plurality of clip circuits is directly connected to the next clip circuit in the series of clip circuits.

40. (Previously Presented) The system of claim 34, wherein the plurality of clip circuits form a pipeline, and each clip circuit is a segment of the pipeline.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.